COLLABORATIVE DATA STREAM MINING IN UBIQUITOUS ENVIRONMENTS USING DYNAMIC CLASSIFIER SELECTION

JOÃO BÁRTOLO GOMES

*Institute for Infocomm Research (I2R), A*STAR, Singapore*
1 Fusionopolis Way Connexis, Singapore 138632

MOHAMED MEDHAT GABER

School of Computing, University of Portsmouth,
Buckingham Building, Lion Terrace, Portsmouth, PO1 3HE,
United Kingdom

PEDRO A. C. SOUSA

Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa,
Quinta da Torre, 2825-114, Caparica,
Portugal

ERNESTINA MENASALVAS *

Facultad de Informática, Universidad Politécnica de Madrid,
Campus de Montegancedo, s/n 28660 Boadilla del Monte, Madrid,
Spain

In ubiquitous data stream mining, different devices often aim to learn concepts that are similar to some extent. In many applications, such as spam filtering or news recommendation, the data stream underlying concept (e.g., interesting mail/news) is likely to change over time. Therefore, the resultant model must be continuously adapted to such changes. This paper presents a novel Collaborative Data Stream Mining (*Coll-Stream*) approach that explores the similarities in the knowledge available from other devices to improve local classification accuracy. *Coll-Stream* integrates the community knowledge using an ensemble method where the classifiers are selected and weighted based on their local accuracy for different partitions of the feature space. We evaluate *Coll-Stream* classification accuracy in situations with concept drift, noise, partition granularity and concept similarity in relation to the local underlying concept. The experimental results show that *Coll-Stream* resultant model achieves stability and accuracy in a variety of situations using both synthetic and real world datasets.

Keywords: Collaborative Data Stream Mining; Ubiquitous Knowledge Discovery; Concept Drift; Performance evaluation

*This research is partially financed by project TIN2008-05924 of Spanish Ministry of Science and Innovation.

1. Introduction and Motivation

The increasing advances and popularity of ubiquitous devices, such as smart phones, PDAs (Personal Digital Assistants) and wireless sensor networks, opens an opportunity to perform intelligent data analysis in such ubiquitous computing environments [14,18].

This work is focused on collaborative data stream mining on-board these ubiquitous devices. The goal is to learn an anytime classification model that represents the underlying concept from a stream of labelled records [21,7]. Such incremental model is used to predict the label of the incoming unlabelled records. However, it is common for the underlying concept of interest to change over time [26,12] and sometimes the labelled data, that is available in the device, is not sufficient to guarantee the quality of the results [19]. Therefore, we propose to use the knowledge available in other devices that is similar to the local underlying concept to collaboratively improve the accuracy of local predictions.

The data mining problem is assumed to be the same in all the devices, however the feature space and the data distributions are not static, as assumed by traditional data mining approaches [15,26]. We are interested in understanding how the knowledge available in other devices can be integrated to improve local predictive accuracy in a ubiquitous data stream mining scenario [18].

As an illustrative example, collaborative spam filtering [4] is one of the possible applications for the proposed collaborative learning approach. Each ubiquitous device learns and maintains a local filter that is incrementally updated from a local data stream based on features extracted from the incoming mails. In addition, the user usage patterns and feedback are used to supervise the filter that represents the target concept (i.e., the distinction between spam and ham). In this scenario, the ubiquitous devices could collaborate by using the knowledge available in the community that is similar to their local concept. Furthermore, the dissemination of knowledge is faster, as devices new to the mining task, or that have access to fewer labelled records, can anticipate spam patterns that were observed in the community, but not yet locally. Moreover, the privacy and computational issues that would result from sharing the original mail are minimised, as only the filters (i.e., models) are shared. Consequently, this has the potential to increase the efficiency of the collaborative learning process. However, such approach has not yet been properly investigated.

Nevertheless, many challenges arise from this collaborative scenario, the two major ones are: i) how the knowledge from the community can be exploited to improve local predictiveness; and ii) how to adapt to changes in the underlying concept. To address these challenges, in this paper, we propose an incremental ensemble approach (*Coll-Stream*) where the models available from the community are selected and weighted based on their local accuracy for different partitions of the feature space. Such technique is motivated by the possible conflicts between models and to capture subspace similarity to the underlying concept. It allows to

exploit the fact that each model can be accurate only for certain subspaces (i.e., where its expertise matches or is similar to the local underlying concept).

Moreover, we performed experiments to study how *Coll-Stream* classification accuracy is influenced by concept drift, noise, partition granularity and concept similarity in relation to the local underlying concept. Our experimental studies show that *Coll-Stream* results in a more stable and accurate incremental model, when compared with state-of-the-art approaches, on a variety of situations using both synthetic and real world datasets.

We should note that the communication costs and protocols to share models between devices are out of the scope of this work and represent an interesting open challenge that we intend to address in future work.

The rest of the paper is organised as follows. The next Section reviews the related work. Section 3 provides the problem definition, which is followed by the description of the *Coll-Stream* approach in Section 4. The experimental setup and results are discussed in Section 5. Finally, in Section 6, this work conclusions and future work are presented.

2. Related work

In collaborative and distributed data mining, the data is distributed and the goal is to apply data mining to different, usually very small and overlapping, subsets of the entire data [5,24]. In this work, our goal is not to learn a global concept, but to exploit the similarities from other devices concepts, while maintaining a local or subjective point of view. Wurst and Morik [30] explore this idea by investigating how communication among peers can enhance the individual local models without aiming at a common global model. The motivation is similar to what is proposed in domain adaptation [6] or transfer learning [20]. Peng et al.[22] propose a fusion approach to provide an optimal ranking of classification models when different multiple criteria decision making (MCDM) methods provide conflicting results. Still, these approaches assume a batch scenario, however, when the mining task is executed in a ubiquitous environment [18], an incremental learning approach is required.

In ubiquitous data stream mining, the feature space of the records that occur in the data stream may change over time [15] or be different among devices [30]. For example, in a stream of documents where each word is a feature, it is impossible to known in advance which words will appear over time, and thus what is the best feature space to represent the documents with. Using a very large vocabulary of words results inefficient, as most of the words will likely be redundant and only a small subset of words is finally useful for classification.

Over time, it is also likely that new important features appear and that previously selected features become less important, which brings change to the subset of relevant features. Such change in the feature space is related to the problem of concept drift, as the target concept may change due to changes in the predictiveness of the available features. However, most existing data stream mining algorithms

are not able to learn from a dynamic feature space and do not explore that some features can be less important to the target concept. Katakis et al. [15] propose the usage of an incremental feature selection to asses feature predictiveness over time and a feature-based classifier that can execute in such dynamic feature space.

This is related with the issue of concept drift [9,2,11] and adaptive modelling [16]. That must also be address in the distributed scenario and is a fundamental difference between our work and the work of Stahl et al. [24]. Moreover, in our previous work we bring awareness to the issue of collaborative learning and describe a similar, but more particular, framework to the one described in this paper. In such work context information must be available [1]. In addition, the work proposed in this paper considers multiple variations of the general framework and includes a thorough experimental study and discussion of how the collaborative approach can bring additional value to ubiquitous knowledge discovery.

Coll-Stream is an ensemble approach to exploit other devices knowledge in a ubiquitous data stream mining scenario [8]. Such techniques have been applied successfully to improve classification accuracy in data mining problems and particularly in data streams, where the underlying concept changes [25,28,17]. The proposed system is most related in terms of the learning algorithm to what has been proposed by Zhu et al. in [32] and Tsymbal et al. in [27] as both approaches consider concept drift, select the best classifier for each record based on its position in the feature space, and are able to learn from data streams. However, in these works the base classifiers are learnt from chunks of a stream of training records in a sequential method. While the classifiers used in *Coll-Stream* are learnt in other ubiquitous devices. Moreover, *Coll-Stream* adapts to concept drift incrementally over time using a time window of fixed size, whereas in the aforementioned works a new classifier is learnt from the next chunk of the stream and an evaluation set is created periodically using the most recent records.

3. Problem Definition

Let X be the space of attributes and its possible values and Y be the set of possible (discrete) class labels. Each ubiquitous device aims to learn the underlying concept from a stream DS of labelled records where the set of class labels Y is fixed. However, the feature space X does not need to be static. Let $X_i = (\vec{x_i}\ ,\ y_i)$ with $x_i \in X$ and $y_i \in Y$, be the i^{th} record in DS. We assume that the underlying concept is a function f that assigns each record x_i to the true class label y_i. This function f can be approximated using a data stream mining algorithm to train a model m at device from the DS labelled records. The model m returns the class label of an unlabelled record $\vec{x}$, such that $m(\vec{x}) = y \in Y$. The aim is to minimise the error of m (i.e., the number of predictions different from f). However, the underlying concept of interest f may change over time and the number of labelled records available for that concept can sometimes be limited. To address such situations, we propose to exploit similarities in models from other devices and use the available labelled

records from DS to obtain the model m. We expect m to be more accurate than using the local labelled records alone when building the model. The incremental learning of m should adapt to changes in the underlying concept and easily integrate new models. We assume that the models from other devices are available and can be integrated at anytime. The costs and methods used to generate and share these models are beyond the scope of this work.

3.1. *Concept Similarity*

The notion of concept similarity followed in this work is based on how the underlying function f agrees/disagrees with other device target function. Since it is not possible to compare the functions directly we compare the degree of agreement between the learnt models. In [31] a measure to compare the similarity between two models is proposed. Given two classification models m_1,m_2 and a sample dataset D_n of n records, it calculates for each record X_i=($\vec{x_i}$, y_i) a score,

$$score(X_i) = \begin{cases} +1 \; if \; m_1(\vec{x}) = m_2(\vec{x_i}) \\ -1 \; if \; m_1(\vec{x}) \neq m_2(\vec{x_i}) \end{cases}$$

that is used to represent the degree of equivalence between m_1 and m_2, that is an average continuous value score with range [-1,1], defined as,

$$ce = \frac{\sum_{X_i \in D_n} score(X_i)}{N}$$

The larger the output value, the higher the degree of conceptual similarity between the models. For the records in D_n it compares how m_1 and m_2 classify the records. The authors [31] argue that the accuracy and the conceptual equivalence degree are not necessarily positively correlated, as models can still achieve the same accuracy and misclassify different parts of the attribute space. Moreover, this approach is independent of the model representation and can be used with heterogeneous models (e.g., decision tree and neural network).

4. Coll-Stream

In this work, we propose *Coll-Stream*, a collaborative learning approach for ubiquitous data stream mining that combines the knowledge of different models from other ubiquitous devices. This collaborative learning process is illustrated in Figure 1.

There is a large number of ensemble methods to combine models, which can be roughly divided into:

- i) voting methods, where the class that gets more votes is chosen [25,28,17];
- ii) selection methods, where the "'best"' model for a particular record is used to predict the class label [32,27,22].

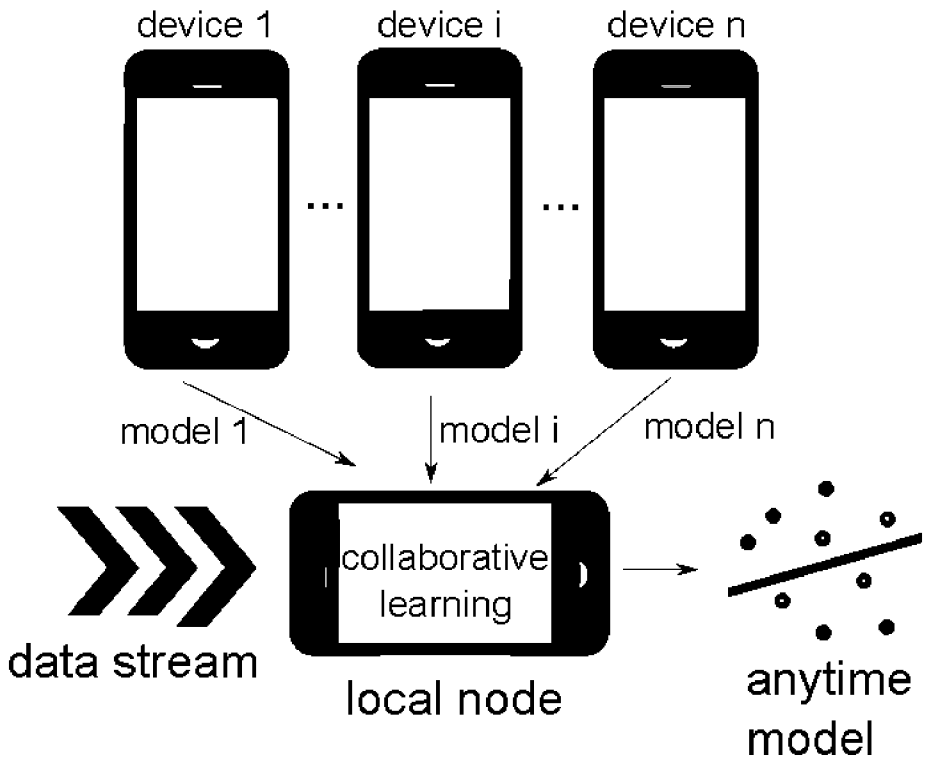

Fig. 1. Collaborative learning process

The *Coll-Stream* is a selection method that partitions the feature space X into a set of regions R. For each region, an estimate of the models accuracy is maintained over a sliding window. This estimated value is updated incrementally as new labelled records are observed in the data stream or new models are available. This process detailed in Algorithm 4.1, can be considered a meta-learning task where we try to learn for each model from the community how it best represents the local underlying concept for a particular region $r_i \in R$. When *Coll-Stream* is asked to label a new record $\vec{x_i}$, the best model prediction is used. The best model is considered to be the one that is more accurate for the partition r_i that contains the new record, as detailed in Algorithm 4.2. The accuracy for a region r_i is the average accuracy for each partition of its attributes. For $r15$ in Figure 2, we average the accuracy for value $V1$ of attribute $A1$ and value $V5$ of attribute $A2$. The accuracy is the number of correct predictions divided by the total number of records observed (these values are updated in lines 10 and 12 of Algorithm 4.1). The next section explains how the regions are created using the attribute values.

Algorithm 4.1 *Coll-Stream* Training

Require: Data stream DS of labelled records, window w of records.

repeat
  Add next record DS_i from DS to w;
  if $w \rightarrow numRecords > wMaxSize$ **then**
    $forget(w \rightarrow oldestRecord)$;
  end if
  $r = getRegion(DS_i)$;
  for all $Model \rightarrow m_j$ **do**
    $prediction := m_j.classify(DS_i)$;
    if $prediction = DS_i \rightarrow class$ **then**
      $updateRegionCorrect(r, m_j)$;
    end if
    $updateRegionTotal(r, m_j)$;
  end for
until END OF STREAM

Algorithm 4.2 *Coll-Stream* Classification

Require: Data stream DS of unlabelled records.

repeat
  Get DS_i from DS;
  $r := getRegion(DS_i)$;
  for all $Model \rightarrow m_j$ **do**
    $model := argmax_j(getAccuracy(m_j, r)$;
  end for
  return $model.classify(DS_i)$;
until END OF STREAM

4.1. *Creating Regions*

An important part of *Coll-Stream* is to learn for each region of the feature space X which model m_j performs better. This way m_j predictions can be used with confidence to classify incoming unlabelled records that belong to that particular region.

The feature space can be partitioned in several ways, here we follow the method used by Zhu et al. [32], where the partitions are created using the different values of each attribute. For example, if an attribute has two values, two estimators of how the classifiers perform for each value are kept. If the attribute is numeric, it is discretised and the regions use the values that result from the discretisation process. This method has shown good results and it represents a natural way of partitioning the feature space. However, there is an increased memory cost associated with a larger number of regions. To minimise this cost the regions can be partitioned into

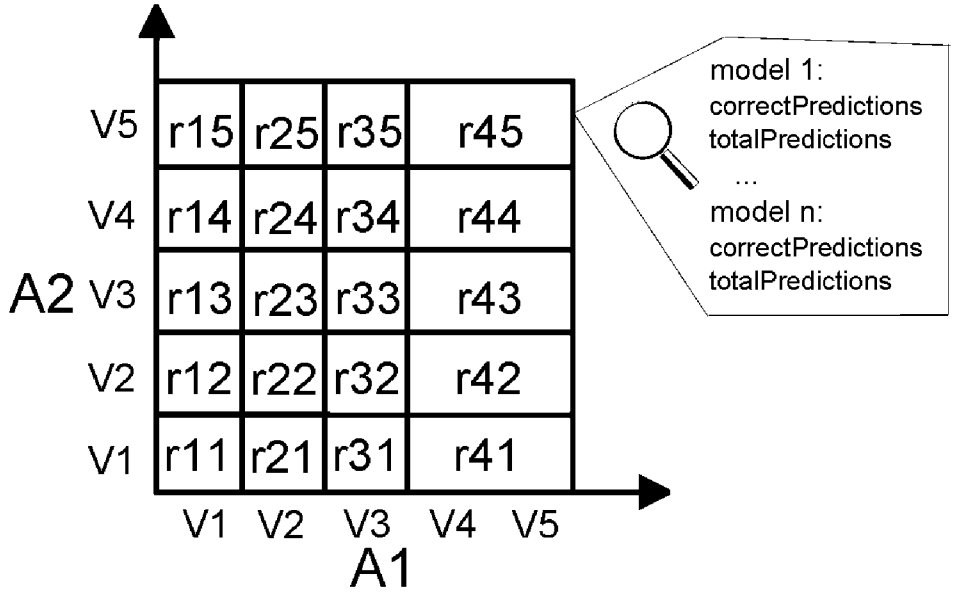

Fig. 2. Partition the feature space into regions

higher granularity ones, aggregating attribute values into a larger partition. This is illustrated in Figure 2, where the values $V4$ and $V5$ of attribute $A1$ are grouped into regions $r41$ to $r45$. In section 5.4, we perform experiments to study how the region's granularity influences the accuracy of the approach.

Figure 3 illustrates the training and classification procedures of *Coll-Stream* that are described in Algorithm 4.1 and Algorithm 4.2.

4.2. *Variations*

Some variations of the *Coll-Stream* approach were considered while developing the method. Details are given in what follows.

4.2.1. *Multiple classifier selection*

If more than one model is selected, their predictions are weighted according to the corresponding accuracy for region r_i, and the record to be labelled gets the class with the highest weighted vote. This is similar to weighted majority voting but with a variable number of classifiers, where the weights are calculated in relation to the region that contains the unlabelled record to classify.

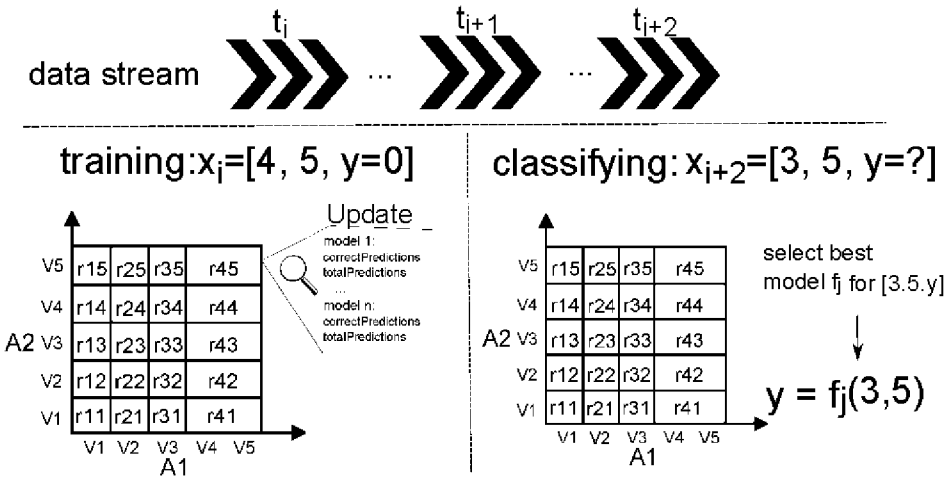

Fig. 3. Coll-Stream: Training and Classifying

4.2.2. *Feature weighting*

The models used from the community can represent a heterogeneous feature space as each one is trained according to a different data stream DS_d. One possible variation is for each device to measure feature relevance. Then at the time of classification the accuracy estimates for each region are weighted according to the feature weight for that region. The predictive score of each feature can be computed using popular methods such as, the information gain, χ^2 or mutual information [15]. However, these must be calculated incrementally given the data stream scenario where this approach is framed. Moreover, this takes into account that features that were relevant in the past can become irrelevant at some point in the future for a different target concept.

4.2.3. *Using local base learner*

One base learner that is trained using the available records in the device can be always part of the ensemble. This way in situations where there is not enough knowledge available from the community, local knowledge can be applied using this classifier. This integration is simple as it only requires an additional step of training the classifier when a new record arrives in addition to updating the ensemble estimates for the new record region.

4.2.4. *Resource awareness*

Resource-awareness is an important issue in ubiquitous data stream mining [8,10]. In such a dynamic ubiquitous usage scenario, it is common for *Coll-Stream* method

to receive too much knowledge from the community over time. In such situations we propose to discard past models that have the lowest performance and allow the integration of new models.

5. Experimental study

We conducted experiments to test the proposed approach accuracy in different situations, using a variety of synthetic and real datasets. The implementation of the proposed learning system was developed in Java, using the MOA [3] environment as a test-bed. MOA [13] stands for Massive Online Analysis and is an open-source framework for data stream mining written in Java. Related to the WEKA project [29], it includes a collection of machine learning algorithms and evaluation tools particular to data stream learning problems. The MOA evaluation features and some of its algorithms were used, both as base classifiers to be integrated in the ensemble and in the experiments for accuracy comparison.

5.1. *Datasets*

A description of the datasets used in our experimental studies is given in the following.

5.1.1. *STAGGER*

This dataset was introduced by Schlimmer and Granger [23] to test the STAGGER concept drift tracking algorithm. The STAGGER concepts are available as a data stream generator in MOA [13] and has been used as a *benchmark* dataset to test concept drift [23]. The dataset represents a simple block world defined by three nominal attributes *size*, *color* and *shape*, each with 3 different values. The target concepts are:

- $size \equiv small \wedge color \equiv red$
- $color \equiv green \vee shape \equiv circular$
- $size \equiv (medium \vee large)$.

5.1.2. *SEA*

The SEA concepts dataset was introduced by Street and Kim [25] to test their Stream Ensemble Algorithm. It is another *benchmark* dataset as it uses different concepts to simulate concept drift, allowing control over the target concepts in our experiments. The dataset has two classes {class0, class1} and three features with values between 0 and 10 but only the first two features are relevant. The target concept function classifies a record as class1 if $f_1 + f_2 \leq \theta$ and otherwise as class0. The features f_1 and f_2 are the two relevant ones and θ is the threshold value between the two classes. Four target concept functions were proposed in [25], using threshold values 8,

9, 7 and 9.5. This dataset is also available in MOA [13] as a data stream generator, and it allows control over the noise in the data stream. The noise is introduced as the $p\%$ of records where the class label is changed.

5.1.3. *Web*

The webKD data set[a] contains web pages of computer science departments of various universities. The corpus contains 4,199 pages (2,803 training pages and 1,396 testing pages), which are categorised into: *project*; *course*; *faculty*; *student*. For our experiments, we created a data stream generator with this dataset and defined 4 concepts, that represent user interest in certain pages. These are:

- $course \vee project$
- $faculty \vee project$
- $course \vee student$
- $faculty \vee student$

5.1.4. *Reuters*

The Reuters dataset[b] is usually used to test text categorisation approaches. It contains 21,578 news documents from the Reuters news agency collected from its newswire in 1987. From the original dataset, two different datasets are usually used, R52 and R8. R52 is the dataset with the 52 most frequent categories, whereas R8 only uses the 8 most frequent categories. The R8 dataset has 5,485 training documents and 2,189 testing documents. In our experiments from R8, we use the most frequent categories: *earn* (2,229 documents), *acq* (3,923 documents) and *others* (a group with the 6 remaining categories, with 1,459 documents). Similar to the Web dataset, in our experiments, we define 4 concepts (i.e., user interest) with these categories. These are:

- $others$
- $earn$
- acq
- $earn \vee others$

5.2. *Experimental Setup*

We test the proposed approach using the previously described datasets with the data stream generator in MOA [13], the target concept was changed sequentially every 1,000 records and the learning period shown in the experiments is of 5,000 records. This number or records allows for each of the concepts to be seen at least once for all the datasets used. In addition, for each concept in all the datasets 1000 records

[b]http://www.cs.umb.edu/ smimarog/textmining/datasets/index.html

is more than required to observe a stable learning curve. As parameters, we used for the window size 100 records and this was fixed for all the experiments and for all the algorithms that use a sliding window. This guarantees the robustness of the approach without fine parameter tuning, which is a drawback of many approaches. The influence of such parameter on the results is contrasted with a version of the $NaiveBayes$ algorithm over a sliding window. Consequently, we can distinguish the gains coming from the collaborative ensemble approach and the ones coming from the adaptation of using a sliding window.

The approaches compared in the experiments are:

- $Coll - Stream$, the approach proposed in this work.
- $MajVote$, Majority Weighted Voting, ensemble approach where each classifier accuracy is incrementally estimated based on its predictions. To classify a record, each classifier votes with a weight proportional to its accuracy. The class with most votes is used.
- $NBayes$, incremental version the *Naive Bayes* algorithm.
- $Window$, incremental Naive Bayes algorithm but its estimators represent information over a sliding window;
- *AdaHoeffNB*, Hoeffding Tree that uses an adaptive window (ADWIN) to monitor the tree branches and replaces them with new branches when their accuracy decreases [2].

In addition, we have implemented and tested the accuracy of $Coll - Stream$ variations proposed in Section 4.2. Nevertheless, the results show only a very small increase in accuracy for the variation that considers the relative importance of the features, and are not significant for the other variations. For this reason the results presented in this section refer to the regular version of $Coll-Stream$ and Section 5.7 is dedicated to describe the experiments of the variation that uses feature selection.

In the experiments, the base classifiers (that represent the community knowledge) used in the ensemble were trained using 1000 data records that correspond to each individual concept.We used the $NaiveBayes$ and *HoeffdingTrees* algorithms available in MOA [13] as base classifiers. Therefore, for each concept the ensemble receives 2 classifiers. For the real datasets, the ensemble only receives the 3 first of the 4 possible concepts, this asserts how the approach is able to adapt the existing knowledge to a new concept that is not similar to the ones available in the community (note that for each still two base classifiers are received). In the experiments we record the average classification accuracy over time using a time window of 50 records using the evaluation features available in MOA [13]. In the synthetic datasets we tested different seeds to introduce variability in the results but because of the large number of records the classifiers easily capture the target concept without the seed causing a significant influence on the accuracy.

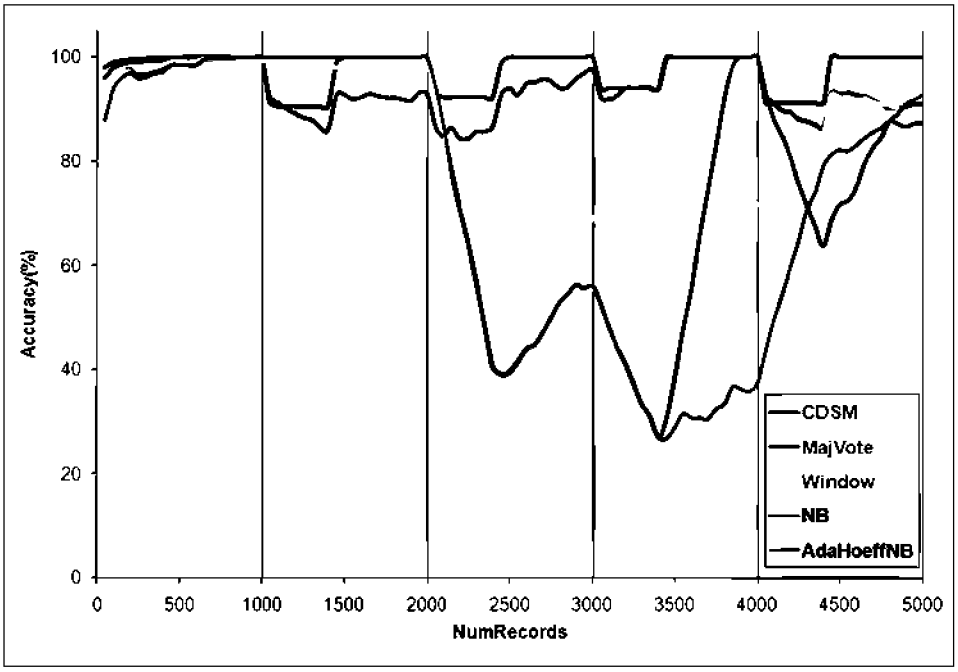

Fig. 4. Accuracy over time for the STAGGER datastream

5.3. *Accuracy evaluation of Coll-Stream*

We compare the efficacy of *Coll-Stream* in relation to the other aforementioned approaches. In this set of experiments we measured the predictive accuracy over time. The vertical lines in the figures indicate a change of concept.

DataSet	STAGGER	SEA	Web	Reuters
AdaHoeffNB	78.86%	89.72%	58.24%	68.08%
NBayes	72.74%	90.96%	57.06%	62.90%
Window	81.96%	92.42%	58.62%	72.36%
MajVote	93.76%	90.98%	66.16%	66.94%
Coll-Stream	**97.42%**	**94.72%**	**71.00%**	**76.92%**

Table 1. Accuracy evaluation

Table 1 shows the overall accuracy of the different approaches for each dataset. Concerning the accuracy, *Coll-Stream* consistently achieves the highest accuracy. For the other approaches, the performance seems to vary across the datasets. The real datasets are very challenging for most of the approaches.

Figure 4 depicts further analysis of the accuracy of the different approaches over

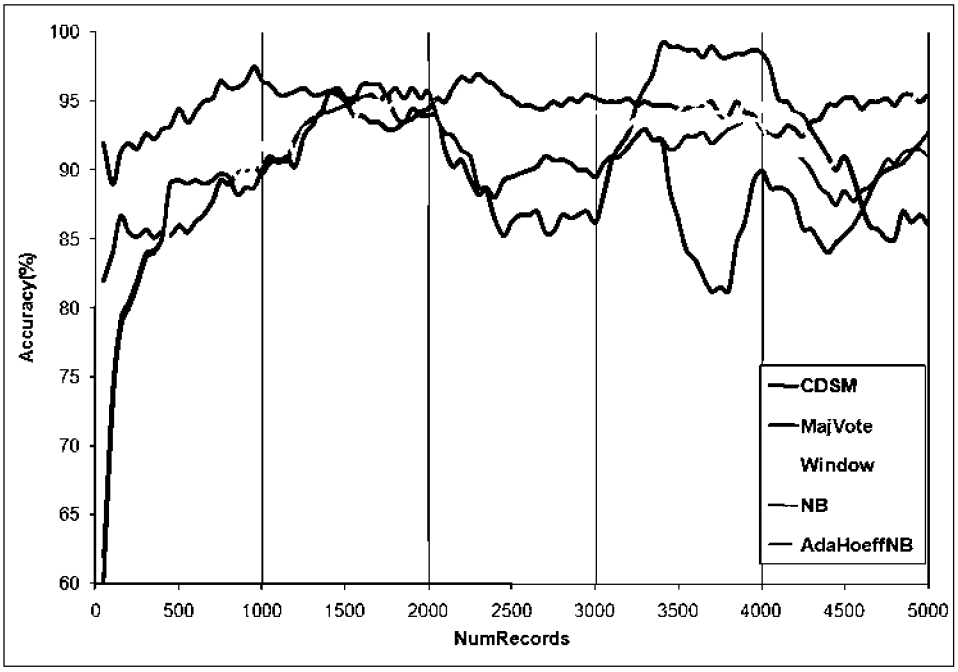

Fig. 5. Accuracy over time for the SEA datastream

time for the $STAGGER$ data. It shows that *Coll-Stream* is not only the more accurate but also the most stable approach, even after concept changes. The $MajVote$ also achieves very good results, close to *Coll-Stream*, but for the 2nd and 3rd concept it performs worse than *Coll-Stream*. For the $Window$, *AdaHoeffNB* and $NBayes$ approaches, the first is able to adapt faster to concept drift, while *AdaHoeffNB* only shows a some gain over the $NBayes$, which is the worst approach in this evaluation, due to the lack of adaptation.

Figure 5 shows the high and stable accuracy of *Coll-Stream* over time for the SEA data. In this experiment, we can observe that the $MajVote$ performs worse than *Coll-Stream*, and do similarly the other methods with the exception of the 4^{th} concept, where the $MajVote$ achieves the best performance. The $Window$ approach also shows good accuracy and stable performance with the changing concept, which makes it higher than $MajVote$ when we look at the overall accuracy in Table 1. The $NBayes$ and *AdaHoeffNB* approaches do not show significant difference. We should note that even the *AdaHoeffNB* which achieves the worse performance in the evaluation is able to keep the accuracy higher than 80%. This can be a result of less abrupt differences between the underlying concepts, when compared with what we observed in the $STAGGER$ data in Figure 4.

The Web data concepts are more complex than the ones that exist in the synthetic data. For this reason, we can observe in Table 1 that the overall accuracy

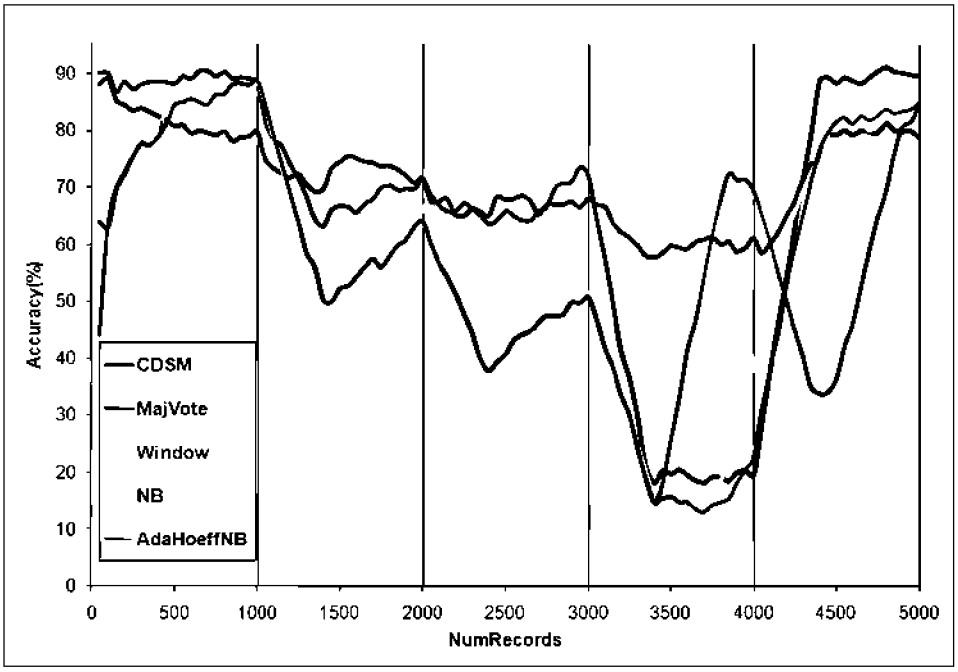

Fig. 6. Accuracy over time for the Web data stream

is not as high for most of the approaches. Figure 6 further analyses the accuracy curve for the different concepts and how it is affected by concept changes. For the 1^{st} concept, the $MajVote$ achieves a slightly better performance than *Coll-Stream*. However, in the 2^{nd} concept we can observe a greater drop in the performance of $MajVote$ at the time that the *Coll-Stream* is more stable and become higher in the accuracy. During the 3^{rd} concept, both approaches achieve similar results, while the other approaches are not able to adapt successfully to the concept changes. It is interesting to find that for the 4^{th} concept, which is dissimilar to the classifiers used in the ensemble approaches, we observe that *Coll-Stream* is able to adapt well with only a slight drop in accuracy while the $MajVote$ shows a large drop in performance and is not able to adapt successfully. Again when the 1^{st} concept recurs we see a dominance of the $MajVote$ which seems to represent this concept with high accuracy.

Using the *Reuters* dataset, the overall accuracy is better than in the Web data as can be observed in Table 1. Figure 7 shows that *Coll-Stream* achieves high accuracy across the concepts and very stable performance over time. The results are somehow similar to the Web ones. However, the $MajVote$ achieves worse performance for the 2^{nd} and 3^{rd} concepts, while the $Window$ approach is able to adapt faster to the different concepts, with the exception of the 3^{rd} one. For the 4^{th} concept, the $Window$ approach is even able to outperform *Coll-Stream*, which explains its overall

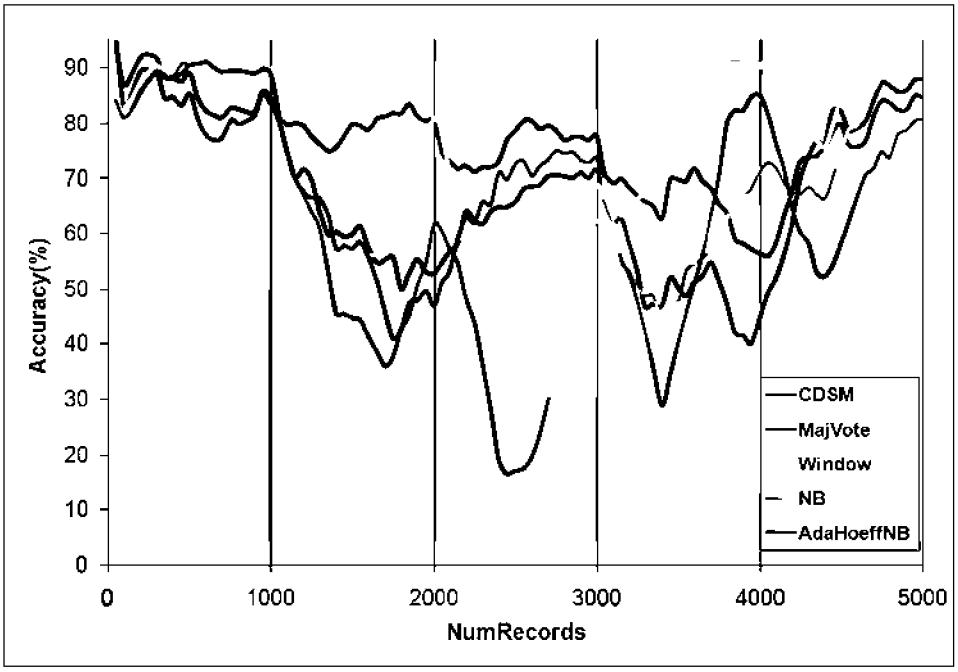

Fig. 7. Accuracy over time for the Reuters datastream

accuracy in Table 1. The *AdaHoeffNB* also is able to adapt to the concept drift but this adaptation is not as fast as *Coll-Stream*.

5.4. *Impact of region granularity on the accuracy*

In this set of experiments we measured how the accuracy is influenced by the granularity of the partitions used in *Coll-Stream*. For the *SEA* dataset where each attribute can take values between 0 and 10. We defined the regions with different sizes from 10 possible values to only 2 values for each attribute. For example, if we consider $R2$, the 2 indicates that each attribute has to be discretised into only two values. Consequently, all the accuracy estimations for attribute values greater than or equal to 5 are stored in one region, while values lower are stored in the another. A similar situation is illustrated in Figure 2 of Section 4.1. Figure 8 shows that the accuracy of *Coll-Stream* decreases with higher region granularity (i.e., less partitions). In Table 2 we measured the memory required for the different granularities and how that size relates to the overall memory consumption of the approach (excluding the classifiers). We observe that the additional memory cost to have higher accuracy is small. This could only have a significant impact in ubiquitous devices with very limited memory where the accuracy-efficiency trade-off of the approach is critical.

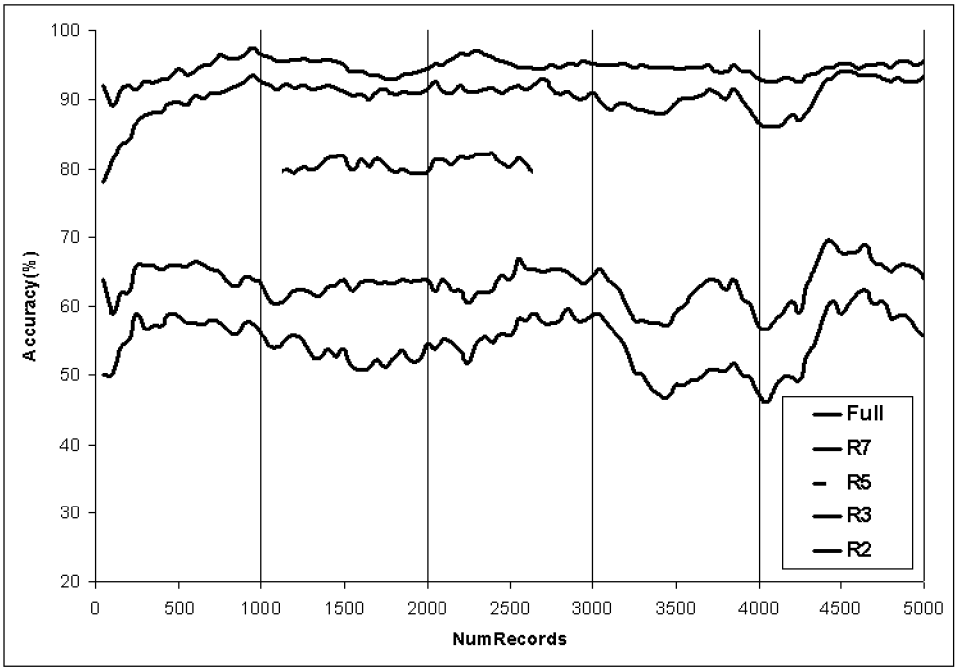

Fig. 8. Accuracy with different region granularity using SEA datastream

Regions	Accuracy	Memory(bytes)	Memory(%)
Full	94.72%	30112	55.83%
R7	90.82%	23776	44.09%
R5	79.58%	20608	38.21%
R3	63.34%	17440	32.33%
R2	54.64%	15856	29.40%

Table 2. Region granularity evaluation using SEA dataset

The results show that *Coll-Stream* can work in situations with memory constraints and still achieve a good trade-off between accuracy and the memory consumed. It can be seen that R5 and R7 are competitive while at the same time saving around 50% memory consumption. The resource efficiency of such approaches to ubiquitous knowledge discovery also opens additional issues for future research work. Particularly, when exploring other representation strategies that can save memory and will also result in lower communication overhead.

DataSet	Noise 0%	Noise 10%	Noise 20%	Noise 30%
AdaHoeffNB	89.72%	81.08%	71.44%	63.12
NBayes	90.96%	81.94%	72.72%	**64.12**
Window	92.42%	82.82%	73.12%	63.90
MajVote	90.98%	81.42%	71.96%	63.90
Coll-Stream	**94.72%**	**83.68%**	**73.22%**	63.78

Table 3. Noise impact evaluation using SEA datastream

5.5. *Impact of noise in the accuracy*

We compare the impact of noise on the accuracy of *Coll-Stream*. Table 3, shows the results of our experiments with different approaches using the *SEA* data with different noise percentages (i.e., percentage of records where the class label changed). The first column represents the case without noise and shows the results that were previously reported in Section 5.3. We can observe that *Coll-Stream* achieves higher accuracy than the other approaches even when the noise level increases, however as the percentage of noise increases the difference between the approaches decreases. Consequently, when the noise level is 30%, all of the approaches achieve a very similar performance (around 63%).

5.6. *Effect of concept similarity in the ensemble*

DataSet	Without TC	With TC
STAGGER	95.86%	97.42%
SEA	94.24%	94.72%
Web	71.00%	72.36%
Reuters	76.92%	77.78%

Table 4. Similarity with target concept (TC)

In Section 5.3, when discussing the evaluation of the experiments using real datasets, we were able to observe (in Figures 6 and 7) that *Coll-Stream* is able to adapt to new concepts that are not represented in the community/ensemble. This is clear when we compare *Coll-Stream* performance difference with the $MajVote$ for the 4^{th} concept (between 4,000 and 5,000 records) in the real datasets. To further investigate this issue, we performed an additional experiment where we measured the impact on the accuracy of *Coll-Stream* when having the target concept represented in the ensemble. We can observe the results in Table 4. The table shows a small drop in the accuracy between the two cases; when the target concept is represented and when it is not. Thus, it could be concluded that *Coll-Stream*

achieves good adaptation to new concepts using existing ones. Furthermore, for the SEA dataset we observe the least difference, because even without knowledge from the 4^{th} concept, there is greater similarity to known ones than in other datasets (e.g., in the $STAGGER$ dataset where the difference between concepts across the regions is greater). Consequently, if there is a local similarity among the concepts, *Coll-Stream* is able to exploit it. This way it can represent a concept by combining other concepts that are locally similar to the target one.

5.7. *Impact of feature selection on the accuracy*

In general, accuracy evaluation of *Coll-Stream* when using feature selection shows that it is possible to maintain or even increase the accuracy while reducing the number of features that need to be kept. This has a strong impact on the accuracy-efficiency trade-off of the approach and will be discussed in detail in the following subsection where we evaluate the memory consumption of *Coll-Stream*. In addition, we observe in Table 5 that there is a small decrease in the accuracy of *Coll-Stream*, particularly in the Web dataset, in this set of experiments in relation to the experiments in the previous section. This is a result of using less diversity in the ensemble (i.e., only models from $NaiveBayes$ as base learner are used)

Table 5 shows the 5 different tested methods their parameters for each dataset and the accuracy obtained. In what concerns the accuracy for the different datasets, we can observe in Table 5 that for the synthetic datasets where the number of features is much smaller than in the real datasets. Therefore, it is only possible to perform a modest reduction on the number of features without affecting the accuracy. This is also a consequence of the number of irrelevant features. For instance, in the $STAGGER$ dataset the number of irrelevant features can be 1 or 2 according to the target concept. Moreover, in the SEA dataset the last feature is always irrelevant to the target concept, we can observe that when the number of kept feature is two (and the feature selection method correctly selects the two predictive ones) the accuracy increases. Nevertheless, if one of the predictive features is lost there is a sharp drop in accuracy.

For the real datasets, where there is a large number of features the results show that its possible to reduce the number of features while achieving a similar or slightly better accuracy than without feature selection.

With respect to the different feature selection methods, either the fixed or threshold approaches achieve similar results. However, the main drawback associated with this method is related to the selection of the appropriate parameter value (i.e., threshold or number of features). In general, the fixed approach allows better control over the consumed space, while the threshold approach is more flexible. This will further analysed in the next section where we asses the memory savings that result from each method.

DataSet	Measure	Fixed(1)	Fixed(2)	Threshold(1)	Threshold(2)	WithoutFS
STAGGER	Accuracy	90.78%	**97.40%**	93.98%	**97.40%**	**97.40%**
	ParValue	1	2	0.13	0.05	-
SEA	Accuracy	87.78%	**96.12%**	95.10%	**96.12%**	94.72%
	ParValue	1	2	0.08	0.06	-
Web	Accuracy	67.38%	66.54%	**67.80%**	66.54%	66.4%
	ParValue	100	300	0.08	0.02	-
Reuters	Accuracy	**77.00%**	76.32%	76.68%	76.06%	75.64%
	ParValue	100	300	0.08	0.02	-

Table 5. Accuracy evaluation of *Coll-Stream* using feature selection

5.8. *Impact of feature selection on memory consumption*

When measuring the savings in memory consumption that result from using feature selection, we can observe in Table 6 that is possible to maintain of even increase the accuracy while consuming at least 50% or less or the resources. Please observe this from the number of features (NumF) and the percentage of memory (Mem) used in relation to the test without feature selection (WihoutFS).

DataSet	Measure	Fixed(1)	Fixed(2)	Threshold(1)	Threshold(2)	WihoutFS
STAGGER	Accuracy	90.78%	**97.40%**	93.98%	**97.40%**	**97.40%**
	NumF	3	6	4	**5**	9
	Memory	33%	66%	44%	**55 %**	100 %
SEA	Accuracy	87.78%	**96.12%**	95.10%	**96.12%**	94.72%
	NumF	4	8	5	8	12
	Memory	33%	66%	42%	66%	100 %
Web	Accuracy	67.38%	66.54%	**67.80%**	66.54%	66.4%
	NumF	300	900	**234**	782	2820
	Memory	14%	43%	**11%**	37%	100 %
Reuters	Accuracy	**77.00%**	76.32%	76.68%	76.06%	75.64%
	NumF	300	900	502	986	1683
	Memory	18%	54%	30	59 %	100 %

Table 6. Memory evaluation of *Coll-Stream* using feature selection

5.9. *Impact of the number of training records on the accuracy*

Finally, we measured how the number of training records in the stream influences the accuracy for the different approaches. We generated the SEA dataset as described previously but with a number of records that ranges from 5000 to 5 million and

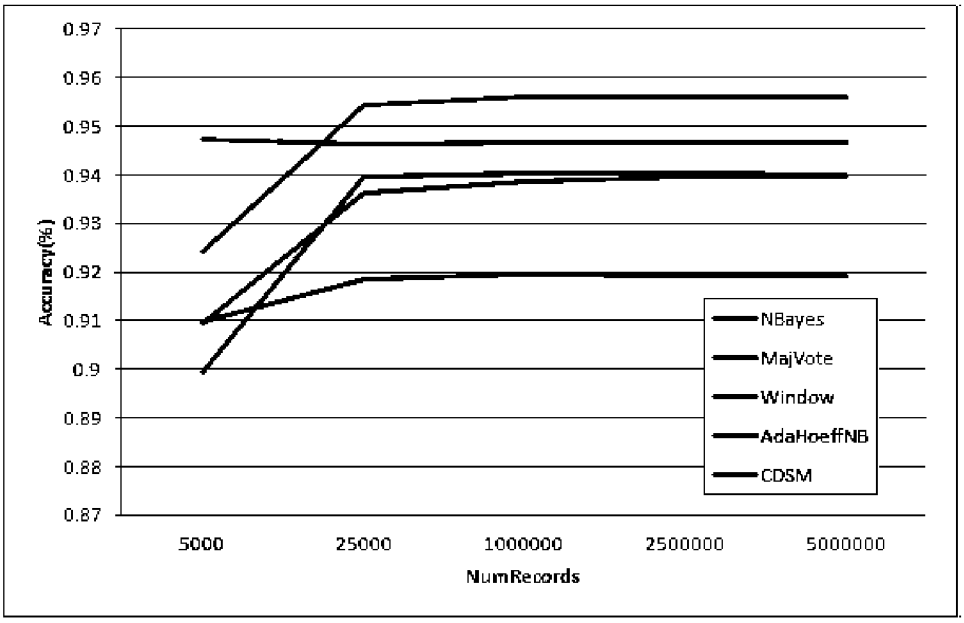

Fig. 9. Accuracy with different number of training records using SEA datastream

measured the overall accuracy. Figure 9 shows the results of our experiment. We can observe that *Coll-Stream* can achieve high accuracy using the least amount of training records. Moreover, it is very stable while other approaches require to process a much higher number of records until the accuracy starts to stabilise. This value is reached around 250.000 records for most methods, with $AdaHoeffNB$ and $Window$ being the approaches that benefit the most from the increased number of records. These results are meaningful for applications where there the number of training records available is limited. We plan to study in future work how to further minimise the need for labelled data exploring semi-supervised and active learning strategies.

6. Conclusions and Future Work

This paper discusses collaborative data stream mining in ubiquitous environments and proposes *Coll-Stream*, an ensemble approach that incrementally learns which classifiers from an ensemble are more accurate for certain regions of the feature space. *Coll-Stream* is able to adapt to changes in the underlying concept using a sliding window of the classifier estimates for each region. Moreover, we also discussed and investigated possible variations of *Coll-Stream*.

In order to evaluate *Coll-Stream*, we developed an implementation of the proposed approach. Several experiments were performed using 2 known datasets for concept drift and 2 popular datasets from text mining from which we create a

stream generator. We tested and compared *Coll-Stream* with other related methods in terms of accuracy, noise, partition granularity and concept similarity in relation to the local underlying concept. The experimental results show that the *Coll-Stream* approach proposed in this paper mostly outperforms the other methods and could be used for situations of collaborative data stream mining as it is able to exploit local knowledge from other concepts that is similar to the new underlying concept.

In future work, we plan to: i) study the communication costs of *Coll-Stream* and investigate efficient protocols to address this problem ii) further explore variations of the approach, for instance if the partitions are not optimal this will negatively influence the accuracy, the dynamic creation of the partitions is an interesting variation to be further explored iii) Exploring semi-supervised and active learning strategies to further minimise the need for labelled data iv) use *Coll-Stream* to support intelligent decision making in a collaborative news recommender application.

Acknowledgments

The work of J.P. Bártolo Gomes was supported by a PhD Grant of the Portuguese Foundation for Science and Technology (FCT) and a mobility grant from Consejo Social of UPM that made possible his stay at the University of Portsmouth. This research is partially financed by project TIN2008-05924 of Spanish Ministry of Science and Innovation. Thanks to the FCT project KDUDS (PTDC/EIA-EIA/98355/2008).

References

1. J. Bártolo Gomes, M. Gaber, P. Sousa, and E. Menasalvas. Context-aware collaborative data stream mining in ubiquitous devices. *Advances in Intelligent Data Analysis X*, pages 22–33, 2011.
2. A. Bifet and R. Gavalda. Adaptive learning from evolving data streams. *Advances in Intelligent Data Analysis VIII*, pages 249–260, 2009.
3. A. Bifet, G. Holmes, R. Kirkby, and B. Pfahringer. Moa: Massive online analysis. *The Journal of Machine Learning Research*, 11:1601–1604, 2010.
4. P. Cortez, C. Lopes, P. Sousa, M. Rocha, and M. Rio. Symbiotic Data Mining for Personalized Spam Filtering. In *Web Intelligence and Intelligent Agent Technologies, 2009. WI-IAT'09. IEEE/WIC/ACM International Joint Conferences on*, volume 1, pages 149–156. IEEE, 2009.
5. S. Datta, K. Bhaduri, C. Giannella, R. Wolff, and H. Kargupta. Distributed data mining in peer-to-peer networks. *IEEE Internet Computing*, pages 18–26, 2006.
6. H. Daumé III and D. Marcu. Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research*, 26(1):101–126, 2006.
7. P. Domingos and G. Hulten. Mining high-speed data streams. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 71–80. ACM New York, NY, USA, 2000.
8. M.M. Gaber, S. Krishnaswamy, and A. Zaslavsky. Ubiquitous data stream mining. In *Current Research and Future Directions Workshop Proceedings held in conjunction with PAKDD*. Citeseer, 2004.
9. Mohamed Medhat Gaber and Philip S. Yu. Detection and classification of changes in

evolving data streams. *International Journal of Information Technology and Decision Making*, 5(4):659–670, 2006.

10. Mohamed Medhat Gaber and Philip S. Yu. A framework for resource-aware knowledge discovery in data streams: a holistic approach with its application to clustering. In *SAC '06: Proceedings of the 2006 ACM symposium on Applied computing*, pages 649–656, New York, NY, USA, 2006. ACM.
11. João Bártolo Gomes, Ernestina Menasalvas, and Pedro AC Sousa. Learning recurring concepts from data streams with a context-aware ensemble. In *Proceedings of the 2011 ACM symposium on applied computing*, pages 994–999. ACM, 2011.
12. João Bártolo Gomes, Pedro AC Sousa, and Ernestina Menasalvas. Tracking recurrent concepts using context. *Intelligent Data Analysis*, 16(5):803–825, 2012.
13. G. Holmes, R. Kirkby, and B. Pfahringer. MOA: Massive Online Analysis, 2007 - http://sourceforge.net/projects/moa-datastream/.
14. H. Kargupta, K. Sarkar, and M. Gilligan. Minefleet®: an overview of a widely adopted distributed vehicle performance data mining system. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 37–46. ACM, 2010.
15. I. Katakis, G. Tsoumakas, and I. Vlahavas. On the utility of incremental feature selection for the classification of textual data streams. *Advances in Informatics*, pages 338–348, 2005.
16. Mihui Kim, Yukyong Jung, and Kijoon Chae. Adaptive data mining approach for flow redirection in multihomed mobile router. *International Journal of Information Technology and Decision Making*, 9(5):737–758, 2010.
17. J.Z. Kolter and M.A. Maloof. Dynamic weighted majority: An ensemble method for drifting concepts. *The Journal of Machine Learning Research*, 8:2755–2790, 2007.
18. S. Krishnaswamy, J. Gama, and M.M. Gaber. Advances in data stream mining for mobile and ubiquitous environments. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 2607–2608. ACM, 2011.
19. Mohammad M. Masud, Jing Gao, Latifur Khan, Jiawei Han, and Bhavani Thuraisingham. A practical approach to classify evolving data streams: Training with limited amount of labeled data. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, pages 929–934, Washington, DC, USA, 2008. IEEE Computer Society.
20. S.J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, pages 1345–1359, 2009.
21. Yi Peng, Gang Kou, Yong Shi, and Zhengxin Chen. A descriptive framework for the field of data mining and knowledge discovery. *International Journal of Information Technology & Decision Making*, 7(04):639–682, 2008.
22. Yi Peng, Gang Kou, Guoxun Wang, and Yong Shi. Famcdm: A fusion approach of mcdm methods to rank multiclass classification algorithms. *Omega*, 39(6):677–689, 2011.
23. J.C. Schlimmer and R. Granger. Beyond incremental processing: Tracking concept drift. In *Proceedings of the Fifth National Conference on Artificial Intelligence*, volume 1, pages 502–507, 1986.
24. F. Stahl, M. Gaber, H. Liu, M. Bramer, and P. Yu. Distributed classification for pocket data mining. *Foundations of Intelligent Systems*, pages 336–345, 2011.
25. W.N. Street and Y.S. Kim. A streaming ensemble algorithm (SEA) for large-scale classification. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 377–382. ACM New York, NY, USA, 2001.

26. A. Tsymbal. The problem of concept drift: definitions and related work. *Computer Science Department, Trinity College Dublin*, 2004.
27. Alexey Tsymbal, Mykola Pechenizkiy, Pádraig Cunningham, and Seppo Puuronen. Dynamic integration of classifiers for handling concept drift. *Inf. Fusion*, 9:56–68, January 2008.
28. H. Wang, W. Fan, P.S. Yu, and J. Han. Mining concept-drifting data streams using ensemble classifiers. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 226–235. ACM New York, NY, USA, 2003.
29. I.H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann Pub, 2005.
30. M. Wurst and K. Morik. Distributed feature extraction in a p2p setting–a case study. *Future Generation Computer Systems*, 23(1):69–75, 2007.
31. Y. Yang, X. Wu, and X. Zhu. Mining in anticipation for concept change: Proactive-reactive prediction in data streams. *Data mining and knowledge discovery*, 13(3):261–289, 2006.
32. Xingquan Zhu, Xindong Wu, and Ying Yang. Effective classification of noisy data streams with attribute-oriented dynamic classifier selection. *Knowl. Inf. Syst.*, 9:339–363, March 2006.